User Guide

Goddard Satellite Data Simulator Unit (G-SDSU)

Version 3.3.3

01 Jun 2014

Authors: Toshi Matsui, Eric Kemp

Table of Contents

1. Overview

Modern multi-sensor satellite observations provide a more complete view of land, cloud, precipitation, and aerosols processes; meanwhile, it is becoming a challenge for remote sensing and modeling communities to harness these observations simultaneously due to inconsistent physics assumptions and spatial scales between satellite retrievals and model physics. To this end, a unified system of multi-sensor simulators, the Goddard Satellite Data Simulator Unit (G-SDSU), has been developed through multi-institutional collaborations [*Matsui et al.* 2013] upon the initial version of HyARC SDSU [*Masunaga et al.,* 2010]. The G-SDSU is the end-to-end satellite simulator, which computes satellite-consistent Level-1 (L1) measurements (e.g., radiance/brightness temperature or backscatter), from outputs of the NASA-Unified Weather Research and Forecasting (NU-WRF) simulation through various simulators:

- Microwave simulator: SSMI, SSMIS, TMI, GMI, AMSR, AMSU, MSU, etc.
- Radar simulator: PR, DPR, CPR, etc.
- Visible IR simulator: AVHRR, MODIS, VIRS, GLI, GOES APR, etc.
- LIDAR simulator: CALIOP, CATS, etc.
- Broadband simulator: ERBE, CERES, etc.

NU-WRF-simulated L1 signals can be directly compared with the satellite-observed L1 signals; therefore G-SDSU bridges model and satellite remote sensing through following paths: i) radiance-based model evaluation and development [*Matsui et al.,* 2009; *Li et al.,* 2010], ii) an operator of radiance-based data assimilation system [*Zupanski et al.,* 2011; *Zhang et al.,* 2013], iii) development of synthetic satellite observations for future satellite missions [*Matsui et al.,* 2013].

1.1 Enhancements

G-SDSU version 3.3.3 was released in June 2014. This release will be along the release of NU-WRF version v7-3.5.1 and include the new features of the NU-WRF v7-3.5.1. Upgrades from the previous version V3.0.0 include i) a new microphysics option for Goddard 4ICE scheme, ii) capability of simulating Doppler velocity (profiler only) in radar simulator, iii) unified treatment of the Goddard class and generalized-gamma classes in microphysics, and iv) bug fixes.

1.2 Current Components

- **Input Model options**: G-SDSU can directly read the NU-WRF's native output format (NetCDF V3), including surface fields, vertical profiles of pressure, geopotential height, water vapor, temperature, vertical velocity, and variety of condensates and aerosol particles. Surface properties include geolocation (latitude and longitude), elevation, land-cover, vegetation fraction, soil temperature, soil moisture, and snow depth.

- **Orbit and Scanning options:** G-SDSU scanning module accounts for satellite orbit and sensor scanning patterns and associated geolocations with respect to NU-WRF model coordinate. Satellite orbit is predicted by Keplerian parameters and Kozai's first-order perturbation parameters. Scanning geolocations can be tracked by scan type, rotation speed and angles, sampling rates. A limited number of satellites and sensors are pre-registered in the G-SDSU. For unregistered instruments, users must provide specific information of such.

- **Cloud Microphysics options**: One/two-moment bulk scheme (e.g., Goddard scheme, RAMS scheme, Morrison scheme) as well as spectra-bin (HUCM SBM scheme) microphysics are supported. Consistent assumption of particle size distributions (PSDs), density, and hydrometeor classes are used to calculate optical properties.

- **Aerosol Microphysics options**: G-SDSU currently supports GOCART aerosol scheme only.

- **Microwave optical properties**: Microwave optical properties (extinction, single scattering albedo, asymmetry, and backscattering coefficient) are computed through the Lorenz-Mie solution with an *a priori* database of the complex refractive indices. Mixture of the refractive indices can be estimated through the Maxwell-Garnet approximation or the effective-medium approximation. Non-spherical single-scattering properties of ice crystals can be optionally obtained from the SCATDB [*Liu,* 2008]; however, database is generally limited to small-size particles. Gaseous extinction and absorption (H_2O, N_2, and O_2) are parameterized by the empirical fitting parameters [*Rosenkranz,* 1993]. These microwave optical properties are computed identically for both the microwave and the radar simulators.

- **Microwave simulator**: Top-of-atmosphere (TOA) emergent microwave brightness temperature (Tb) is derived from two-stream radiative transfer with the Eddington's second approximation along the model column [*Kummerow,* 1993]. More realistic pseudo 3D (1D slant radiance path) calculation [*Olson et al.,* 2006] is available with microwave scanning option, which account for satellite orbit and scanning along the microwave simulator. Horizontal and vertical polarization is accounted from surface properties only. If horizontal grid spacing of the NU-WRF simulation is finer than the footprint size of microwave, simulated microwave Tb is convoluted within effective field of view (EFOV). There is an option that computes the bottom-of-atmosphere (BOA) downwelling microwave Tb for a ground-based radiometer.

- **Radar simulator**: Attenuated and non-attenuated radar reflectivity and Doppler velocity profiles are derived through a single scattering process along the model column [*Masunaga and Kummerow,* 2005]. More realistic pseudo 3D (1D slant radiance path) calculation [*Meneghini and Kozu,* 1990] is available with radar scanning option, which accounts for satellite orbit and instrumental scanning along the radar simulation and surface clutter impact. If horizontal grid spacing of the NU-WRF simulation is finer than the footprint size of microwave, simulated reflectivity are convolved for range resolutions and instantaneous field of view (IFOV). There is an option that computes radar reflectivity from ground.

- **Visible IR optical properties**: Visible-IR optical properties (extinction, single scattering albedo, phase function, and backscattering coefficient) are computed through the Lorenz-Mie kernel with an *a priori* database of the complex refractive indices of clouds and aerosols. Gaseous extinction and absorption (H_2O, CO_2, O_3, N_2O, CO, CH_4, and O_2) are parameterized by the k-distribution method using the HITRAN 2004 database [*Sekiguchi and Nakajima,* 2008]. Since NU-WRF, without WRF-CHEM, generally predicts only water vapor (H_2O), vertical profiles of various gaseous constituents are interpolated at the model pressure levels from the climatological values. These visible-IR optical properties are treated identically between the visible-IR simulator and the lidar simulator.

- **Visible IR simulator**: TOA emergent visible radiance and IR Tb are derived from a discrete-ordinate radiative transfer scheme along the model column [*Nakajima and Tanaka,* 1986] with different phase-function calculation modes [*Nakajima and Tanaka,* 1988]. The number of stream can modulate accuracy of multiple scattering processes; the default is six streams that result in reasonable convergence of accuracy in treating multiple scattering processes. Because of this, visible-IR simulator takes several times more computational time than microwave simulator.

- **Lidar simulator**: Attenuated and non-attenuated lidar backscattering profiles are computed from single scattering process along the model column [*Platt,* 1973]. Backscatter is computed at each range resolution. Since lidar instruments have finer horizontal resolution IFOV than the NU-WRF simulation, there is no backscatter convolution process. Multiple scattering and instrumental random noises are not currently calculated in the lidar module.

- **Broadband optical properties**: Broadband optical properties (extinction, single scattering albedo, and asymmetry parameter) are computed from line-by-line integration using the visible-IR optical properties. Molecular absorptions are parameterized by correlated K method, and have not been integrated between broadband and visible-IR optical properties yet.

- **Broadband simulator**: TOA Broadband fluxes and profiles of radiative heating rate are computed through the delta-Eddington two-stream adding approximation [*Chou and Suarez,* 1999; *Chou et al.,* 2001]. Shortwave scheme accounts for the absorption due to water vapor, O_3, O_2, CO_2, clouds, and aerosols. Longwave scheme accounts for the absorption due to

major gases absorption and most of the minor trace gases, as well as clouds and aerosols. Accuracy of the parameterized broadband fluxes is within 1% in comparison with the high spectral-resolution line-by-line calculations.

- **Surface Properties**: Either a dynamic or a static module computes over-land microwave surface emissivity. The dynamic module is the CRTM microwave emissivity model [*Weng et al.,* 2001], while the static module is TELSEM [*Aires et al.,* 2011]. For both scheme, at strong-rainfall pixel, surface is treated as water body that mimics surface inundation. Over-ocean microwave emissivity is a function of ocean salinity, temperature, and wind speed [*Wilheit,* 1979]. Visible surface albedo is derived from MODIS satellite product, and stored mean values of different surface land-cover type in look-up tables. There is no consideration of bi-directional reflectance distribution function (BRDF) over land. Over ocean, visible-IR albedo/emissivity is calculated through matrix formulation that accounts for BRDF in response to sensor-solar cone angle and wind speed effect.

- **GV simulator**: This simulator is specifically designed for GPM Ground-Validation instruments: aircraft- (*2D-P* and *2D-C*) and surface-based (*Parsivel*) microphysics probe. Currently, it has been only applied to HUCM spectra-bin microphysics scheme [*Iguchi et al.,* 2012a; *Iguchi et al.,* 2012b]. GV module integrate and re-sample HUCM SBM bin-by-bin microphysics information, and generate aircraft 2D- or ground Parsivel-observable bulk microphysics properties.

- **MPI**: The G-SDSU can optionally utilize Message Passing Interface (MPI) library [] for parallel computation in a multi-core to super computers. Two MPI options are available. First option is file-number decomposition, which has greater advantage to process a large number of model inputs. Second option is domain decomposition, which assign different tiles of NU-WRF domains for different processors. The second option also involves memory decomposition; thus it has advantageous to process a large-domain model file (or complex bin microphysics).

1.3 Updates

- **Goddard 4ICE microphysics option**: Goddard 4ICE microphysics scheme [Lang et al. 2013, submitted to JAS] is most recent version of the Goddard bulk single-moment microphysics scheme. It incorporated a hail class in addition to the existing hydrometeor classes (cloud, rain, ice, snow aggregate, and graupel). Snow aggregate has diagnostic size distributions as a function of temperature. Effective snow density depends on the size close to the aircraft observation. Graupel has also diagnostic size distributions. Rain and hail classes have exponential size distribution with fixed intercepts. G-SDSU accounts for all of these assumptions of 4ICE, when converting them into optical properties.

- **Doppler velocity in radar simulator**: In radar simulator, Doppler velocity is also computed in addition to the existing radar reflectivity output. Doppler velocity is calculated by

reflectivity-weighted averaging terminal velocity of hydrometeors with background updraft velocity (omega in NU-WRF). Currently, Doppler velocity is computed as “profiler” model, which mean normal direction to the surface, and does simulate “weather radar” types of instruments.

2. Using the Software

2.1 Structure of the Source Code

G-SDSU V3.3.3 is included within the package of NU-WRF V7-3.5.1. After opening NU-WRF package, you will find G-SDSU directory, there are following sub-directories.

`SRC: Source codes of G-SDSU (Fortran.F programs).`

`QRUN_****: Default directory (**** could be anything) to run G-SDSU.`

`INPUTS: Default directory to store model (NU-WRF) input files.`

`OUTPUTS: Default directory to store the simulated satellite signals.`

`DATAFILES: various run-time input files.`

`SSLUT: Default directory for storing single-scattering look-up tables.`

`REFFERENCE: References for G-SDSU.`

`QUICK_START_GSDSU.txt is the quick-start guide.`

2.2 Source Code Structure and Building

G-SDSU is built upon object-oriented Fortran90 programming (and some old Fortran codes) and C programming. C-preprocessor is used for MPI library and their call routines. Dynamic allocation allows better memory management for large memory data. Many computational intensive calculations use pre-computed look-up tables that can boost the computational speed (> 10x) without loosing measure accuracy. The G-SDSU can optionally utilize Message Passing Interface (MPI) library for parallel simulations in a multi-core to super computers. Thus, compilation of the G-SDSU requires 1) Fortran compiler, 2) MPI library (optional), 3) NetCDF library, 4) C-compiler, 5) C-Preprocessor, and 6) Make utility in the standard Unix-flavor machine (UNIX, LINUX, Mac OSX, and etc.). Thus, users of G-SDSU require basic knowledge of these computing environments.

For building the G-SDSU,

1. Go to SRC directory.

2. Modify `makefile` according to the compiler and library options.

- Make sure QRUN directory (running directory) must be consistent to what you defined.

```
QRUN     = ./../QRUN_***       (*** is unique name)
```

- Modify compiler options, library, and flags. (Some parameters are available for the NCCS Discover and the NAS Pleiades as default.) Binary data is assumed in little endian format (in case of Intel Fortran `-assume byterecl`), and you must put appropriate compiler flag for this.

```
CPP      = /lib/cpp -C -w
CF       = ifort
CFFLAGS = -O3 -ip -assume byterecl
CC       = icc
INC_NETCDF = /nasa/netcdf/3.6.0/intel/include
LD_NETCDF = /nasa/netcdf/3.6.0/intel/lib
LINK_MPI = -lmpi
```

3. Modify define_CPP.h file depending on your library (HDF or MPI).

- MPI option can be either MPI 0 - no mpi, 1 - file decomposition, or 2- domain decomposition. Note that MPI=1 option is most efficient to process many input files, but it often creates memory issues with large-domain inputs. File MPI=2 option is more memory friendly, since it assigns sub-domain on each CPU memory.

```
# define MPI 2
```

- HDF option is made for synthetic GPM simulator, and generally not used: 0 - no HDF library, 1 - HDF library.

```
# define HDF 0
```

4. Compile the source codes by typing
>make

This will take about ~10 seconds to a few minutes, depending on the Fortran compiler flags. Executable (G.SDSU.x) will be copied in QRUN_#### directory.

2.3 Running G-SDSU for Checking IO

For the first time, it is important to check G-SDSU running without any IO problems.

1. Make necessary changes for run-time parameters in Configure_SDSU.F to have consistent IO directory. Only change the direction and do not change any other options at this point.

Inactivate all simulator switches for the 1st time (we just want to check IO process).

```
$simulator_switch
  micro  = .false.      ! Microwave simulator switch; on when .true.
```

```
  radar  = .false.     ! radar simulator switch; on  when .true.
  visir  = .false.     ! visible/IR simulator switch; on when .true.
  lidar  = .false.     ! Lidar simulator switch; on when .true.
  broad  = .false.     ! Broad-band simulator switch; on when .true.
  GV     = .false.      ! GV simulator switch; on when .true.
$end
```

Here is the default setting of io_option for first-time running.

```
$io_options
  sdsu_dir_sslut= './../SSLUT/'  ! directory for the single-scattering LUTs
  sdsu_dir_data = './../DATAFILES/ ! directory for various datafiles
  sdsu_io_name  = 'inpfile' ! name of model-input-list file
  verbose_SDSU  = .false.   ! if true, print out more comments during run.
  write_surface = .false.   ! if true, write out surface properties
  write_opt     = .false.   ! if true, write out single scattering properties
  write_CRM3D   = .true.   ! if true, write out CRM 3D file in GrADS format
  write_CRM2D   = .true.   ! if true, write out CRM 2D file in GrADS format
  write_grads_ctl = .true.  ! if true, write grads control files (logical)
  write_psdmmt  = .false.   ! if true, write out PSD moments from HUCM SBM
  output_suffix = '_test'   !it can be any, if you don't want just black ''

  minlat        =  58.0  ! min longitude of lat-lon output ( -90 ~  90) [deg]
  maxlat        =  62.0  ! max latitude  of lat-lon output ( -90 ~  90) [deg]
  minlon        =  21.0  ! min longitude of lat-lon output (-180 ~ 180) [deg]
  maxlon        =  28.0  ! max longitude of lat-lon output (-180 ~ 180) [deg]
  res_latlon    =  0.05  ! grid spacing  of lat-lon output [deg]

$end
```

`write_CRM3D` and `write_CRM2D` are generally activated for diagnostic purpose. In general, `write_grads_ctl` must be always activated so that G-SDSU will write GrADS control files. `minlat`, `maxlat`, `minlon`, and `maxlon` are specific latitude-longitude output domain, and they should be close to the input NU-WRF domain. `res_latlon` could be close to the resolution of satellite footprint.

Input NU-WRF information must be specified within

```
$crm_options
 …….
 ……
 ….
$end
```

User must specify correct information of `sdsu_dir_input`, `sdsu_dir_output`, `mxgridx`, `mxgridy`, `mxlyr`, `gridsize`. For processing NU-WRF information, `sim_case` must be always `'WRF'`. Note that `mxgridx` and `mxgridy` can be smaller than actual maximum dimension of the NU-WRF file, but it cannot be larger than that.

```
  sim_case  = 'WRF'  ! NU-WRF (character*10)
  sdsu_dir_input  = './../INPUTS/' ! G-SDSU input (character*200)
  sdsu_dir_output = './../OUTPUTS/'  ! output directory (character*200)
```

```
  mxgridx   = 423    ! max grid # in horizontal x direction  (integer)
  mxgridy   = 411    ! max grid # in horizontal y direction (integer)
  mxlyr     = 60     ! max grid # in vertical direction  (integer)
  gridsize  = 6.e0   ! horizontal grid spacing [km] (real)
```

Make sure microphysics options are consistent to your NU-WRF simulation.

```
cloud_microphysics = 'GMP_4ICE'  ! Cloud Microphysics Type (character*20)
                                 ! GOD: Goddard 1-mmt scheme [Tao et al. 2003]
                                 ! GOD10: Goddard scheme 2010 [Lang et al. 2010]
                                 ! GMP_4ICE: Goddard 4ICE scheme [Lang et al.
2013]
                                 ! LIN: LIN bulk 1-mmt scheme  [Lin et al. ]
                                 ! WSM: WRF-Single-Moment 6-Class Scheme [Hong et
al. 2004]
                                 ! RAMS1: RAMS 1-mmt scheme [Walko et al., 1995]
                                 ! RAMS2: RAMS 2-mmt scheme [Meyers et al., 1997]
                                 ! HUCM_SBM: HUCM spectra-bin microphysics scheme
[Khain et al. 2007]
                                 ! HUCM_SBM43: HUCM spectra-bin microphysics 43
bin scheme [Khain et al. 2010]
                                 ! MORR: Morrison two-moment scheme.

 gsfc_hail = .false.             ! GSFC hail options (graupel --> hail)
```

Below is special setting for idealized forward simulations. For NU-WRF case, users won't activate these parameters.

```
clear_sky_scene = .false.   ! if .true., zero out all condensates (cloud-
precip) to create clear sky.

 uniform_surface = .false.              ! When it is true, this option assigns
spatially uniform
                                        ! surface characters over the entire
domain.
                                        ! (When sim_case='GCE', this must be
always .true., because GCE input
                                        !  does not have these surface
parameters.)

 idealized_surface%lat       = -12.75e0 ! latitude [deg]
 idealized_surface%lon       = 131.5e0  ! lon      [deg]
 idealized_surface%frac_veg  =   0.e0   ! vegetation fraction [%] (optional
for WRF)
 idealized_surface%albedo    =   0.05   ! surface SW albedo [-]
 idealized_surface%h2o_snow  =   0.e0   ! snow water equivalent [kg m-2]
 idealized_surface%h2o_soil  =   0.9e0  ! soil moisture fraction [0-1]
 idealized_surface%elev      =   0.e0   ! surface elevation [m]
 idealized_surface%dhgt_snow =   0.e0   ! snow depth [m]
 idealized_surface%iland     =   2      ! 1-land,  2-water
 idealized_surface%igbp_typ  =   0      ! IGBP land-cover type (dominant
vegetation type )
```

If you are running GOCART aerosols with WRF-CHEM, you must activate `account_aerosol` in order to read GOCART aerosol mass concentrations.

```
account_aerosol   = .false.    ! if true, account aerosol particles (logical)
```

2. Prepare input-file-list `inpfile`, which is a simple text file that contains a list of input file names. This can be easily done by going to your NU-WRF output directory, and using the command

>ls wrf*d02* > inpfile

This means that all WRF domain 2 output will be written in inpfile.

Your `inpfile` looks like,

> vi inpfile

```
wrfout_d02_2010-09-21_07:00:00
wrfout_d02_2010-09-21_08:00:00
```

Then, copy inpfile into your QRUN_#### directory.

>cp inpfile $GSDSU/QRUN_####

3. Running G-SDSU executable.

Go to QRUN_*** directly first.

a. Running on your desk-top machine,

- For the single CPU option, simply type

  >./GSDSU.x

- For the multiple CPU (MPI) option,

  >mpirun -n## GSDSU.x

b. Running in Discover or Pleiades Super computer.

- For the single CPU option, simply type

  >./GSDSU.x

- For the multiple CPU (MPI) option, you must submit Q_script:

  >sbatch Sbatch***.sh (for Discover)

>qsub Qbatch***.sh (for Pleiades)

Before submitting Q script, you must modify Q script (Qbatch***.sh) for your own setting and resource requirements. Details can be seen in NCCS website (http://www.nccs.nasa.gov/primer/computing.html).

```
#PBS -S /bin/csh
#PBS -N gsdsu
#PBS -l select=7:ncpus=9:model=wes
#PBS -l walltime=00:10:00
#PBS -W group_list=s0809
#PBS -m abe
#PBS -q debug
#PBS -V
#PBS -e /nobackupp1/tmatsui/SDSU/QRUN_LPVEX_DPR
#PBS -o /nobackupp1/tmatsui/SDSU/QRUN_LPVEX_DPR/sdsu_run_mpi_2.txt
```

c. Determining the number of CPUs in MPI runs.

- For MPI=1 (file decomposition) options, MPI will subdivide the number of input file into the discrete number of CPUs. For example, with 24 input files (e.g., NU-WRF netcdf output files), if you set 24 (2) CPU, each CPU will process 1 (12) file. Thus, it can be any number between 1 and the total input number (N), however, it would be ideal that the CPU number can divide N without residuals. E.g., if N=24, the number of CPUs could be 2, 3, 4, 6, 8, 12.

- For MPI=2 (domain decomposition) options, MPI will subdivide the input domain (i-j-k domain) into small tiles (di-dj-k domains). The CPU number should be any integer that can divide the domain without residual (equal sub-domain size). For example, with 10x10 (ixj) domain, the CPU number should be 2 (two 5x10 subdomains), 4 (four 5x5 subdomains), 5 (five 2x10 subdomains), 10 (10 1x10 subdomain), and so on. These number can be derived from running the Fortran executable
  > ./HOW_MANY_CPU_GSDSU
  This asks your domain size, and provides the combination of node number and total number of CPUs.

Initial configuration, all simulators are inactivated (.false.) so that G-SDSU essentially read NU-WRF netcdf files for checking IO process. You must make sure all of IO process works right. If there are problems, it could be due to wrong IO information in the Configure_SDSU.F file.

2.4 Configuration of Running Satellite Simulator

Once you successfully run G-SDSU for initial IO checking, you are ready to run G-SDSU for simulating satellite signals.

2.4.1 Running Microwave Simulator

Activate microwave simulator switch.

```
$simulator_switch
  micro  = .true.     ! microwave simulator switch; on when .true.
  radar  = .false.     ! radar simulator switch; on  when .true.
  visir  = .false.     ! visible/IR simulator switch; on when .true.
  lidar  = .false.     ! Lidar simulator switch; on when .true.
  broad  = .false.     ! Broad-band simulator switch; on when .true.
  GV     = .false.     ! GV simulator switch; on when .true.
$end
```

Options of microwave simulator is configured within

```
$micro_options
……
……
……
……
$end
```

There are two type of microwave simulator configuration.

a. Column Simulator (traditional): This option compute radiative transfer along the model column (normal to surface) at every single NU-WRF grid with fixed sensor view angle, and simulated microwave Tb is convolved with simple convolution techniques that mimic Gaussian weighting. Users must provide following information (example of AMSR-E).

```
micro_sensor = 'AMSR-E' !sensor name (AMSR-E)
ground_micro = .false.  !=.true. for ground based; =.false. for satellite
based
inc_angle_micro = 55.0   ! incident angle [deg]
mxfreq_micro = 6     ! The number of microwave-radiometer channels
freq_micro =   6.925, 10.65, 18.7, 23.8, 36.5, 89.0  ! Channel frequencies
[GHz]
nch_micro = '6.925G', '10.65G', '18.7G ', '23.8G ', '36.5G ' , '89.0G '  !
lut character that is consistent to freq
fov_ct_micro =  43.0, 30.0, 16.0, 18.0,  8.0, 4.0       ! Spatial resolution
for cross-track FOV
fov_dt_micro =  74.0, 51.0, 27.0, 31.0, 14.0, 6.0       ! Spatial resolution
for down-track FOV
```

There are a number of different sensors specified in the configure file, which are masked out by "!" at the beginning of the raw.

b. Scanning Simulator (advanced): This option computes radiative transfer along satellite positions and sensor-pointing surface geolocations through rigorous orbit-scanning simulations. FOV-convolution is based on half-power beam-width angles with Gaussian gain functions including side lobe impact. Currently, only pre-registered satellite and sensors can be chosen from the configuration files. Any combination of the satellite type and sensor type technically works, but it should be consistent to the existing combination (e.g., TRMM_POST and TMI, DMSP_F17 and SSMIS, or Aqua and AMSR).

```
 scan_micro = .true.  ! If true, scan / orbit simulator will run. (logical)
                      ! If true, specify satellite/instrument name below.
                      ! if true, all of above column setups are ignored.

 satellite_micro = 'GPM'    ! Available satellite type (character*20)
                            ! GPM: GPM core satellite
                            ! TRMM_PRE: TRMM satellite before orbit boost
                            ! TRMM_POST: TRMM satellite after orbit boost
                            ! AQUA: Aqua satellite
                            ! DMSP_F16: DMSP satellite F16
                            ! DMSP_F17: DMSP satellite F17

 scan_type_micro = 'GMI_LF'  ! Available sensor scan type
 ! GMI_LF : GMI low-frequency channels  (10.65, 18.7, 23.8, 36.5, 89.0GHz)
 ! GMI_HF : GMI high-Frequency channels (166, 183pm1, 183pm3, 183pm6GHz)
 ! TMI_LF : TMI low-frequency channels  (10.65, 19.35, 21.3, 37.0GHz)
 ! TMI_HF : TMI high-Frequency channels (85.5GHz)
 ! AMSR_E_LF : AMSR-E low-freq channels  (6.925, 10.65, 18.7, 23.8, 36.5)
 ! AMSR_E_HF : AMSR-E high-freq channels (89.0 GHz)
 ! GMI_LF37 : just test purpose
 ! SSMIS : SSMIS imager channels (19.35, 37, 91.655, 150, 183pm3,
183pm6.6GHz)

 inpfile_overpass_micro  = 'overpass_gmi_lf'  ! name of model-input-list file
(character)
```

It is important to set up `inpfile_overpass_micro`. You must prepare this file similar to `inpfile`, but contains satellite overpassing information.

e.g.,

> vi overpass_gmi_lf

```
wrfout_d02_2010-09-21_07:00:00 -30.5  150.0    A     TAG01234
wrfout_d02_2010-09-21_08:00:00 -32.0  160.2    D     TAG05432
```

1st column is NU-WRF file name, consistent to `inpfile`.
2nd column is satellite overpassing latitude (-90 ~ 90deg)
3rd column is satellite overpassing longitude (-180 ~ 180deg)
4rh column is "A" or "D", which is ascending or descending overpass, respectively.
5th column is satellite-overpassing tag, which may be obtained from actual satellite L1B file or you can specify whatever you want.

Satellite overpassing latitude and longitude can be derived from the existing satellite orbit data (available from satellite L1B data products), or they can be any latitude and longitude for synthetic database as long as within the NU-WRF domain.

Once successfully run with the scan_microwave option, G-SDSU provides a following set of outputs in your specified output directory `sdsu_dir_output`.

This binary (and GrADS ctl file) output file contains satellite overpass information in a global lat-lon map.

```
wrfout_d02_2006-08-06_01:00:00.FOV_AQUA_AMSR_E_HF_test.bin
wrfout_d02_2006-08-06_01:00:00.FOV_AQUA_AMSR_E_HF_test.ctl
```

This NetCDF output file contains partial-orbit simulated orbital and sensor geolocation information as well as simulated microwave Tb only over the NU-WRF domain. This is closest to satellite L1B orbital data format.

```
wrfout_d02_2006-08-06_01:00:00.AMSR_E_HF.ORBITAL.AMMA_test.nc
```

This NetCDF output file contains full-orbit simulated orbital and sensor geolocation information.

```
wrfout_d02_2006-08-06_01:00:00.AMSR_E_HF.GEOLOCATION.AMMA_test.nc
```

This binary (and GrADS ctl file) output file contains microwave Tb just over the NU-WRF domain in a regional lat-lon (specified in `minlat`, `maxlat`, `minlon`, and `maxlon`) map.

```
wrfout_d02_2006-08-06_01:00:00.AMSR_E_HF.ORBITAL.AMMA_test.latlon.bin
wrfout_d02_2006-08-06_01:00:00.AMSR_E_HF.ORBITAL.AMMA_test.latlon.ctl
```

2.4.2 Running Radar Simulator

Activate radar simulator switch.

```
$simulator_switch
  micro  = .false.      ! microwave simulator switch; on when .true.
  radar  = .true.      ! radar simulator switch; on  when .true.
  visir  = .false.      ! visible/IR simulator switch; on when .true.
  lidar  = .false.      ! Lidar simulator switch; on when .true.
  broad  = .false.      ! Broad-band simulator switch; on when .true.
  GV     = .false.       ! GV simulator switch; on when .true.
$end
```

Options of radar simulator can be configured within

```
$radar_options
……
……
……
……
$end
```

There are two types of radar simulator configuration.

a. Column Simulator (traditional): This option computes radiative transfer along the model column (normal to surface) at every single NU-WRF grid with fixed sensor view angle, and simulated radar reflectivity and Doppler velocity are convolved with simple convolution technique that mimics Gaussian weighting. Users must provide following parameters (example of TRMM PR).

```
radar_sensor = 'PR'       !sensor name (TRMM PR) (character*20)
ground_radar = .false.    !=.true. for ground-based sensor; =.false. for
satellite-based sensor (logical
mxfreq_radar = 1          !The number of channels (integer)
min_echo = 17.            !minimal_detactable echo [dBZ]
inc_angle_radar = 12.13   !incident angle [deg]
k2 = 0.925         !Radar constant |k^2| defaults (if not known -> -999.)
freq_radar = 13.8   !Channel frequencies [GHz]
nch_radar = '13.8G'  !lut character that is consistent to freq_radar
fov_ct_radar = 5.0  !Spatial resolution for cross-track FOV
fov_dt_radar = 5.0 !Spatial resolution for down-track FOV
mxhgt_radar  = 15.0 ! maximum height of measurement (above sea level) [km]
range_radar = 0.25 ! radar measurement range resolution [km]
```

There are a number of different sensors specified in the configure file, which are masked out by "!" at the beginning of the row.

b. Scanning Simulator (advanced): This option computes radiative transfer along satellite positions and sensor-pointing surface geolocations through rigorous orbit-scanning simulations. FOV-convolution is based on half-power beam-width angles with Gaussian gain functions including side lobe impact. Currently, only pre-registered satellite and sensors can be chosen from the configuration files. Any combination of the satellite type and sensor type technically works, but it should be consistent with the existing combination (e.g., CLOUDSAT and CPR, TRMM_POST and PR, or GPM and DPR_Ku).

```
scan_radar = .false.  ! If true, scan / orbit simulator will run.
                      ! Slow but more accurate geometry. (logical)
                      ! If true, specify satellite/instrument name below.
                      ! if true, all of above sensor setups are ignored.

satellite_radar = 'CLOUDSAT' ! Available satellite type (character*20)
                      ! GPM: GPM core satellite (2013~)
                      ! TRMM_PRE: TRMM satellite before orbit boost
                      ! TRMM_POST: TRMM satellite after orbit boost
                      ! CLOUDSAT: CloudSat satellite

scan_type_radar = 'CPR'  ! Available sensor scan type
                      ! DPR_Ku : GPM Dual-Frequency Radar Ku band (13.6GHz)
                      ! DPR_Ka : GPM Dual Frequency Radar Ka band (35.5GHz)
                      ! PR     : TRMM Precipitation Radar (13.6GHz)
                      ! CPR    : CloudSat Cloud Profiling Radar (94GHz)
```

```
inpfile_overpass_radar  = 'overpass_cloudsat'  ! name of model-input-list
file  (character)
```

It is important to set up `inpfile_overpass_radar`. You must prepare this file similar to `inpfile_overpass_microwave` (Section 2.4.1).

Once successfully run with the scan_radar option, G-SDSU provide following sets of output in your specified output directory `sdsu_dir_output`.

This binary (and GrADS ctl file) output file contains satellite overpass information in a global lat-lon map.

```
wrfout_d02_2006-08-06_01:00:00.FOV_CLOUDSAT_CPR.bin
wrfout_d02_2006-08-06_01:00:00.FOV_CLOUDSAT_CPR.ctl
```

This NetCDF file contains partial-orbit simulated sensor geolocation information as well as CPR reflectivity just over the NU-WRF domain. This is closest to satellite L1B data format.

```
wrfout_d02_2006-08-06_01:00:00.CPR.ORBITAL.AMMA.nc
```

This NetCDF file contains full-orbit simulated sensor geolocation information over the globe.

```
wrfout_d02_2006-08-06_01:00:00.CPR.GEOLOCATION.AMMA.nc
```

This binary (and GrADS ctl file) output file contains partial-orbit reflectivity profiles and sensor geolocations just over the NU-WRF domain.

```
wrfout_d02_2006-08-06_01:00:00.CPR.ORBITAL.AMMA.bin
wrfout_d02_2006-08-06_01:00:00.CPR.ORBITAL.AMMA.ctl
```

This special statistical (GrADS ctl file) output file is only for CPR. It is joint echo-top and echo-depth diagram which mimic ISCCP diagram.

```
wrfout_d02_2006-08-06_01:00:00.CPR.ISCCP.AMMA.bin
wrfout_d02_2006-08-06_01:00:00.CPR.ISCCP.AMMA.ctl
```

2.4.3 Running Visible-IR Simulator

Activate visible-IR simulator switch.

```
$simulator_switch
  micro  = .false.      ! microwave simulator switch; on when .true.
  radar  = .false.      ! radar simulator switch; on  when .true.
  visir  = .true.      ! visible/IR simulator switch; on when .true.
  lidar  = .false.      ! Lidar simulator switch; on when .true.
  broad  = .false.      ! Broad-band simulator switch; on when .true.
  GV     = .false.       ! GV simulator switch; on when .true.
$end
```

Options of visible-IR simulator can be configured within

```
$visir_options
……
……
……
……
$end
```

There are two types of visible-IR simulator configuration.

a. Column Simulator (traditional): This option computes radiative transfer along the model column (normal to surface) at every single NU-WRF grid with fixed sensor view angle, and simulated radiance and/or infrared Tb are convolved with a simple convolution technique that mimic Gaussian weighting. However, convolution may not be used, because visible-IR sensors tend to have higher resolution IFOV than NU-WRF grids. Users must provide following information (example of TRMM VIRS).

```
visir_sensor = 'VIRS'   !sensor name  (TRMM VIRS)
znth_slr = 0.e0    ! solar zenith angle [deg] (if -999. coszen depends on
model time.)
znth_obs = 12.13   ! viewing zenith angle [deg]
azmth    = 40.     ! azimuth angle between the sun and sensor [deg]
mxwavel = 5        !The number of channels
wavel = 0.62, 1.6, 3.8, 10.8, 12.0 !Channel wavelengths  [micron]
nch_wavel =
'0.62micron','1.6micron','3.8micron','10.8micron','12.0micron' !lut character
consistent to wavel
fov_ct_visir =  1.,1.,1.,1.,1.    !Spatial resolution for cross-track FOV
fov_dt_visir =  1.,1.,1.,1.,1.    !Spatial resolution for down-track FOV
```

There are also number of different sensors specified in the configure file, which is masked out by "`!`" at the beginning of the raw.

b. Scanning Simulator (advanced): This option computes radiative transfer along NU-WRF grid column, but swath width and sensor viewing angle are specified for each grid through rigorous orbit-scanning simulations. Convolution is not used, because usually visible-IR sensor has higher resolution than NU-WRF simulation. Because input parameters are complex, only pre-registered satellite and sensors can be chosen from the configuration files. So far, the list of sensors is very limited due to a lack of sensor mechanics information. If `scan_visir = .false.`, traditional column simulator will run.

```
scan_visir = .true.  ! If true, scan / orbit simulator will run to account
more accurate scan geometry.(logical)
                       ! If true, specify satellite/instrument name below.
                       ! if true, all of above sensor setups are ignored.

satellite_visir = 'AQUA'    ! Available satellite type (character*20)
                              ! AQUA: Aqua satellite (2002 May ~)
                              ! TERRA: Terra satellite (2000 Jan ~ )
```

```
                                ! TRMM_PRE: TRMM satellite before orbit boost
                                ! TRMM_POST: TRMM satellite after orbit boost

scan_type_visir = 'MODIS_IR'   ! Available sensor scan type
                                ! MODIS_IR : MODIS sensor IR channel (11um )

inpfile_overpass_visir  = 'overpass_amma_aqua'  ! name of model-input-list
file  (character)
```

It is important to set up `inpfile_overpass_visir`. You must prepare this file similar to `inpfile_overpass_microwave` (Section 2.4.1).

After a successful run, G-SDSU provides a following set of outputs in your specified output directory `sdsu_dir_output`.

This binary (and GrADS ctl file) output file contains satellite overpass information in a global lat-lon map.

```
wrfout_d02_2006-08-06_01:00:00.FOV_MODIS_IR_test.bin
wrfout_d02_2006-08-06_01:00:00.FOV_ MODIS_IR_test.ctl
```

This NetCDF output file contains partial-orbit simulated sensor geolocation information as well as visible-IR radiance just over the NU-WRF domain. This is closest to satellite L1B data format.

```
wrfout_d02_2006-08-06_01:00:00.MODIS_IR.ORBITAL.AMMA_test.nc
```

This NetCDF output file contains full-orbit simulated orbital and sensor geolocation information over the globe.

```
wrfout_d02_2006-08-06_01:00:00.MODIS_IR.GEOLOCATION.AMMA_test.nc
```

This binary (and GrADS ctl file) output file contains simulated IR Tb just over the NU-WRF domain in regional lat-lon (specified in `minlat`, `maxlat`, `minlon`, and `maxlon`) map.

```
wrfout_d02_2006-08-06_01:00:00.MODIS_IR.ORBITAL.AMMA_test.latlon.bin
wrfout_d02_2006-08-06_01:00:00.MODIS_IR.ORBITAL.AMMA_test.latlon.ctl
```

2.4.4 Running Lidar Simulator

Activate lidar simulator switch.

```
$simulator_switch
  micro  = .false.     ! microwave simulator switch; on when .true.
  radar  = .false.     ! radar simulator switch; on  when .true.
  visir  = .false.     ! visible/IR simulator switch; on when .true.
  lidar  = .true.     ! Lidar simulator switch; on when .true.
  broad  = .false.     ! Broad-band simulator switch; on when .true.
  GV     = .false.      ! GV simulator switch; on when .true.
$end
```

Options of lidar simulator can be configured within

```
$lidar_options
……
```

……
……
……
$end

There are two types of lidar simulator configuration.

a. Column Simulator (traditional): This option computes radiative transfer along the model column (normal to surface) at every single NU-WRF grid at specified range resolution. Convolution is not available, because lidar sensors usually have higher resolution than NU-WRF grid. Users must provide following information (example of CALIOP).

```
lidar_sensor = 'CALIOP'          ! sensor name in three characters
ground_lidar = .false.           !=.true. for ground-based sensor; =.false.
for satellite-based sensor (logical)
MS_Correct = 0.7                 ! multiple scattering correction factor
(real)
mxwavel_lidar = 1                ! The number of channels (integer)
wavel_lidar = 0.532              ! Channel wavelengths  [micron]
nch_wavel_lidar = '0.532micron' ! lut character that is consistent to wavel
inst_profile_lidar = .true.      ! = .true. for instrument-defined profile
mxhgt_lidar  =  20.0             ! maximum height of measurement (above sea
level) [km]
range_lidar  =  0.3              ! lidar measurement range resolution [km]
```

b. Scanning Simulator (advanced): This option computes radiative transfer along NU-WRF column at the satellite orbit track through rigorous orbit-scanning simulations. Thus, computational speed may be faster than traditional approach. So far, the list of sensor is very limited due to lack of sensor mechanics information. If `scan_lidar = .false.`, traditional column simulator will be run.

```
scan_lidar = .true.  ! If true, scan / orbit simulator will run to account
more accurate scan geometry.(logical)
                       ! If true, specify satellite/instrument name below.
                       ! if true, all of above sensor setups are ignored.

satellite_lidar = 'CALIPSO'    ! Available satellite type (character*20)
                              ! CALIPSO: CALIPSO satellite

scan_type_lidar = 'CALIOP'  ! Available sensor scan type
                            ! CALIOP : CALIOP (532nm, 1064nm)

inpfile_overpass_lidar  = 'overpass_amma_calipso'  ! name of model-input-list
file  (character)
```

It is important to set up `inpfile_overpass_lidar`. You must prepare this file similar to `inpfile_overpass_microwave` (Section 2.4.1).

After a successful run, G-SDSU provides a following set of outputs in your specified output directory `sdsu_dir_output`.

This binary (and GrADS ctl file) output file contains satellite overpass information in a global lat-lon map.

```
wrfout_d02_2006-08-06_01:00:00.FOV_CALIPSO_CALIOP.bin
wrfout_d02_2006-08-06_01:00:00.FOV_CALIPSO_CALIOP.ctl
```

This binary (and GrADS ctl file) file contains lidar backscatter profile just over the NU-WRF domain.

```
wrfout_d02_2006-08-06_01:00:00.CALIOP.ORBITAL.AMMA.bin
wrfout_d02_2006-08-06_01:00:00.CALIOP.ORBITAL.AMMA.ctl
```

Backscatter-color ratio joint PDF diagram in GrADS format are created.

```
wrfout_d02_2006-08-06_01:00:00.CALIOP.Joint_PDF.AMMA.bin
wrfout_d02_2006-08-06_01:00:00.CALIOP.Joint_PDF.AMMA.ctl
```

2.4.5 Running Broadband Simulator

Activate microwave simulator switch.

```
$simulator_switch
  micro  = .false.      ! microwave simulator switch; on when .true.
  radar  = .false.      ! radar simulator switch; on  when .true.
  visir  = .false.      ! visible/IR simulator switch; on when .true.
  lidar  = .false.      ! Lidar simulator switch; on when .true.
  broad  = .true.      ! Broad-band simulator switch; on when .true.
  GV     = .false.       ! GV simulator switch; on when .true.
$end
```

Options of broadband simulator is configured within

```
$broad_options
……
……
……
……
$end
```

There are two types of broadband simulator configuration.

a. Column Simulator (traditional): This option compute radiative transfer along the model column (normal to surface) at every single NU-WRF grid with fixed sensor view angle, and simulated energy budgets and radiative heating rate profiles are convolved with simple convolution technique that mimics Gaussian weighting. Users must provide the following information.

```
broad_scheme = 'goddard'   ! goddard - Goddard Radition (CliRad) scheme

heating_rate = .false. ! write out 3D broadband SW/LW heating rate [K/day]
                       ! in addition to the default energy budget output.
```

```
fov_ct_broad =  10.e0    ! Spatial resolution for cross-track FOV (CERES)
fov_dt_broad =  10.e0    ! Spatial resolution for down-track FOV  (CERES)
```

b. Scanning Simulator (advanced): Currently, this option is not available for broadband simulator.

2.4.6 Running GV Simulator

GV simulator is only designed for WRF-SBM, a special version of the WRF with HUCM SBM scheme, which has not yet been incorporated to NU-WRF. Following option can generate aircraft- or ground-based prove-observable microphysics parameters from the WRF-SBM simulations.

Activate GV simulator switch.

```
$simulator_switch
  micro  = .false.     ! microwave simulator switch; on when .true.
  radar  = .false.     ! radar simulator switch; on  when .true.
  visir  = .false.     ! visible/IR simulator switch; on when .true.
  lidar  = .false.     ! Lidar simulator switch; on when .true.
  broad  = .false.     ! Broad-band simulator switch; on when .true.
  GV     = .true.      ! GV simulator switch; on when .true.
$end
```

Specify following parameters:

```
particle_shape = 1  ! 0 - sphere
                    ! 1 - irregular (assumption from SnowFake and 2DVD measurements)

aircraft_on = .true.  ! true - simulate aircraft 2D-probe measurable parameters,
                        ! then dump output file (***.GV3D.bin), including          following
parameters.
                        ! bulk water content [g/m3]
                        ! bulk effective radius [micron]
                        ! bulk particle volume [cm3/m3]
                        ! bulk density [g/cm3]
                        ! liquid water fraction [-]

aircraft_ice = .true. ! if true, it only sample ice particle
                        ! if .false. it only sample liquid particle

parsivel_on = .true.  ! true - simulate ground-based Persivel measurable parameters
                        ! , then dump output (***.GV2D.bin), including following parameters
                        ! Parsivel 5min rainfall accumulation [mm]
                        ! Geonor 5min rainfall accumulation [mm]
                        ! bulk effective density [g/cm3]
                        ! bulk effective radius [micron]

parsivel_liq_cutoff = .true. !if True, Parsivel parameters are only accounted for diameter less
than 8mm.


  dump_psd = .false.   ! true - output full 33-bin PSD (***.GV3D_PSD.bin or ***.GV2D_PSD.bin).

  zonal_sampling_on = .false. ! true - sample Aircraft- or PARSIVEL-simulator parameters in
specific zone
```

```
                              ! and dump zonal statistical output (mean normalized PSDs)

                             ! if zonal_sampling_on is true, define sampling zone below
  min_lat = 0.e0             ! minimum latitude [deg] (Aircraf and Parsivel)
  max_lat = 90.e0            ! maximum latitude [deg] (Aircraf and Parsivel)
  min_lon = -180.e0          ! minimum longitude [deg] (Aircraf and Parsivel)
  max_lon =  0.e0            ! maximum longitude [deg] (Aircraf and Parsivel)
  min_alt = 0.e0             ! minimum altitude [km]
  max_alt = 5.e0             ! maximum altitude [km]
```

2.4.7 Additional Settings

With each simulator setting, Configure_SDSU.F has the following options for single-scattering properties (mostly microwave optical properties). Default settings are recommended unless you understand physics or want to investigate sensitivity/uncertainties of forward simulations.

```
$single_scatter_options

  lut_micro    = .true.    ! Particle single-scattering LUT options for micro/radar simulator
(logical).
                           !.true.  : Use LUTs for microwave option - Very Fast
                           !.false. : Full solution of Mie routine. Slow, but accurate.
                           ! (This must be false for HUCM_SBM or HUCM_SBM43 microphysics.)

  lut_visir    = .true.     ! Particle single-scattering LUT option for visir simulator (logical)
                            !.true.  : Use LUTs for microwave opt. Very fast.
                            !.false. : Full solution of Mie routine Slow, but accurate.
                            ! (This must be .false. for HUCM_SBM or HUCM_SBM43 microphysics.)

  lut_replace = .false.     ! Replace existing mie LUT, if you modify single-scattering routines
(logical).
                           !.true.  : Replace single-scattering LUTs.
                           !.false. : Use existing Mie LUTs data.

  ss_opt_micro = 1        ! single scattering calculation options for Microwave/Radar simulator
(integer)
                          ! 1 - Mie      (flufy sphere)
                          ! 2 - SCATDB   (non-spherical ice crystals)
                          ! 3 - SnowFake (non-spherical ice crystals) , not available

  scatdb_ice_type = 0     ! SCATDB ice crystal shape index (if ss_opt_micro = 2, you must choose)
                          ! 0-hexl, 1-hexs, 2-hexb, 3-hexf, 4-hexp, 5-ros3,
                          ! 6-ros4, 7-ros5, 8-ros6, 9-sstr, 10-sden

  ice_refraction_func = 1 ! Effective refraction functions for frozen particles for
Microwave/Radar simulator (integer)
                          ! 1: Oblique Maxwell-Garnett function that assumes ice inclusion within
air matrix.
                          ! 2: Oblique Maxwell-Garnett function that assumes air inclusion within
ice matrix.
                          ! 3: Effective-Medium function that assumes homogeneous mixing.

  melt_opt = 2   ! Effective refraction functions for melting particles for Microwave/Radar
simulator (integer)
                 ! 0: Does not account melting particle
                 ! 1: Oblique Maxwell-Garnett function that assumes ice inclusion within water
matrix.
                 ! 2: Oblique Maxwell-Garnett function that assumes water inclusion within ice
matrix.
                 ! 3: Oblique Maxwell-Garnett function averaging option 1 and 2 --> RECOMMENDED
                 ! 4: Effective-Medium function that assumes homogeneous mixing.


  land_emiss_micro =  2   ! Land-surface emissivity scheme
                          ! 0: simple scheme
                          ! 1: NESDIS land emissivity model (V1)
```

```
                          ! 2: TELSEM emissivity database
$end
```

2.5 Known Issues and Limitations

- G-SDSU is designed for meso-scale meteorological modeling. Thus, model horizontal grid spacing should be less than ~5km.

- It should be able to run multiple simulators simultaneously. However, it requires more computational memory and often causes unusual crash in the middle of the simulations.

- Orbital GPM DPR simulation requires large memory due to the size of orbital output. When you submit MPI run through Q script in Pleiades or Disocover machines, it would be safe to request a half number of total CPU (e.g., Westmere has 12 CPUs total in one node. You may request only 6 CPUs per node.)

- CloudSat CPR and CALIPSO CALIOP simulations do not include multiple-scattering effect. Users must be careful when investigating backscattering signals within optically thick targets.

3. Citations

Aires, F., Prigent, C., and Bernardo…, F. (2011), A Tool to Estimate Land-Surface Emissivities at Microwave frequencies (TELSEM) for use in numerical weather prediction, Quarterly Journal of the Royal Meteorological Society, Volume 137, Issue 656, pages 690–699.

Aoyama, Y., and J. Nakano (1999), *Rs/6000 sp: Practical MPI programming*, IBM Poughkeepsie, New York.

Chou, M.-D., and Suarez, M. J. (1999), A solar radiation parameterization for atmospheric studies, *NASA Tech. Memo*, *10460*40.

Chou, M.-D., Suarez, M. J., Liang, X.-Z., and Yan, M. M.-H. (2001), A thermal infrared radiation parameterization for atmospheric studies, *NASA Tech. Memo*, *19*56.

Iguchi, T., Matsui, T., Shi, J. J., Tao, W., Khain, A. P., Hou, A., Cifelli, R., Heymsfield, A., and Tokay, A. (2012a), Numerical analysis using WRF-SBM for the cloud microphysical structures in the C3VP field campaign: Impacts of supercooled droplets and resultant riming on snow microphysics, *Journal of Geophysical Research: Atmospheres (1984–2012)*, *117*(D23).

Iguchi, T., Matsui, T., Tokay, A., Kollias, P., and Tao, W. (2012b), Two distinct modes in one-day rainfall event during MC3E field campaign: Analyses of disdrometer observations and WRF-SBM simulation, *Geophysical Research Letters*, *39*(24).

Kummerow, C. (1993), On the accuracy of the Eddington Approximation for radiative transfer in the microwave frequencies, *J. Geophys. Res.*, *98*2757-2765.

Li, X., Tao, W.-K., Matsui, T., Liu, C., and Masunaga, H. (2010), Improving a spectral bin microphysical scheme using long-term TRMM satellite observations, *Quart. J. Roy. Meteor. Soc*, *136*382-399.

Liu, G. (2008), A database of microwave single-scattering properties for nonspherical ice particles, *Bulletin of the American Meteorological Society*, *89*(10), 1563-1570.

Masunaga, H., and Kummerow, C. (2005), Combined radar and radiometer analysis of precipitation profiles for a parametric retrieval algorithm, *J. Atmos. Ocean. Tech.*, *22*909-929.

Masunaga, H., Matsui, T., Tao, W.-K., Hou, A. Y., Kummerow, C. D., Nakajima, T., Bauer, P., Olson, W. S., Sekiguchi, M., and Nakajima, T. Y. (2010), Satellite data simulator unit: A multisensor, multispectral satellite simulator package, *Bulletin of the American Meteorological Society*, *91*(12), 1625-1632.

Matsui, T., et al. (2013), GPM Satellite Simulator Over Ground Validation Sites, *Bulletin of the American Meterological Socienty*, (in press).

Matsui, T., Zeng, X., Tao, W.-K., Masunaga, H., Olson, W. S., and Lang, S. (2009), Evaluation of long-term cloud-resolving model simulations using satellite radiance observations and multifrequency satellite simulators, *Journal of Atmospheric and Oceanic Technology*, *26*(7), 1261-1274.

Meneghini, R., and Kozu, T. (1990), Spaceborne weather radar, *Norwood, MA, Artech House, 1990, 208 p.*, *1*.

Nakajima, T., and Tanaka, M. (1986), Matrix formulations for the transfer of solar radiation in a plane-parallel scattering atmosphere, *Journal of Quantitative Spectroscopy and Radiative Transfer*, *35*(1), 13-21.

Nakajima, T., and Tanaka, M. (1988), Algorithms for radiative intensity calculations in moderately thick atmospheres using a truncation approximation, *Journal of Quantitative Spectroscopy and Radiative Transfer*, *40*(1), 51-69.

Olson, W. S., Kummerow, C. D., Yang, S., Petty, G. W., Tao, W.-K., Bell, T. L., Braun, S. A., Wang, Y., Lang, S. E., and Johnson, D. E. (2006), Precipitation and latent heating distributions from satellite passive microwave radiometry. Part I: Improved method and uncertainties, *Journal of applied meteorology and climatology*, *45*(5), 702-720.

Platt, C. (1973), Lidar and radioinetric observations of cirrus clouds, *Journal of the atmospheric sciences*, *30*(6), 1191-1204.

Rosenkranz, P. W. (1993), Absorption of microwaves by atmospheric gases, *Atmospheric remote sensing by microwave radiometry(A 95-14701 02-46), New York, NY, John Wiley & Sons, Inc., 1993,* 37-90.

Sekiguchi, M., and Nakajima, T. (2008), A k-distribution-based radiation code and its computational optimization for an atmospheric general circulation model, *Journal of Quantitative Spectroscopy and Radiative Transfer*, *109*(17), 2779-2793.

Weng, F., Yan, B., and Grody, N. C. (2001), A microwave land emissivity model, *Journal of Geophysical Research: Atmospheres (1984–2012)*, *106*(D17), 20115-20123.

Wilheit, T. T. (1979), A model for the microwave emissivity of the ocean's surface as a function of wind speed, *Geoscience Electronics, IEEE Transactions on*, *17*(4), 244-249.

Zhang, S. Q., Zupanski, M., Hou, A. Y., Lin, X., and Cheung, S. H. (2013), Assimilation of precipitation-affected radiances in a cloud-resolving WRF ensemble data assimilation system, *Monthly Weather Review*, *141*(2), 754-772.

Zupanski, D., Zhang, S. Q., Zupanski, M., Hou, A. Y., and Cheung, S. H. (2011), A prototype WRF-based ensemble data assimilation system for dynamically downscaling satellite precipitation observations, *Journal of Hydrometeorology*, *12*(1), 118-134.